



WINDOWS SERVER 2012 JA AD- HALLINTA POWERSHELLILLÄ

Iiro Koivisto

Opinnäytetyö
Kesäkuu 2014
Tietotekniikan
koulutusohjelma
Tietoliikennetekniikka ja
tietoverkot

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Tietoliikennetekniikka ja tietoverkot

IIRO KOIVISTO:

Windows Server 2012 ja AD-hallinta PowerShellillä

Opinnäytetyö 26 sivua
Kesäkuu 2014

Opinnäytetyön tavoitteena oli selvittää PowerShellin käyttömahdollisuuksia Microsoftin Windows Server 2012 –palvelinkäyttöjärjestelmän ja aktiivihakemiston hallinnassa. Palvelimen ja aktiivihakemiston hallinta työssä on pääosin pyritty suorittamaan muodostamalla etäyhteys toimialueen ulkopuolisesta koneesta palvelimelle.

PowerShell on Microsoftin komentorivipohjainen työkalu Windows – käyttöjärjestelmien hallintaan. Sen tehtävänä on helpottaa ja tehostaa järjestelmänhallintaa paikallisesti ja etänä. Työssä esitellään yleisesti PowerShell – komentotyökalu, perehdytään työkalun historiaan, käyttöliittymään, käyttökohteisiin, ISE –skriptienhallintatyökaluun ja tärkeimpiin komentoihin.

Windows Server 2012 on Microsoftin kuudes julkaisu kehittämästään palvelinkäyttöjärjestelmästä. Työssä on esitelty palvelimen keskeisimmät ominaisuudet, historia, roolit ja käyttöliittymät. Lisäksi perehdyttiin aktiivihakemiston käyttöön ja ominaisuuksiin.

Käytännön osuudessa asennettiin palvelimeen Windows Server 2012 –käyttöjärjestelmä ja muodostettiin siihen etäyhteys PowerShellillä. Palvelinroolien hallinnan ja toimialueen luomisen lisäksi osuudessa käsiteltiin organisaatioyksiköiden ja käyttäjätilien hallintaa aktiivihakemistossa.

Opinnäytetyön tuloksena todettiin Powershellin olevan hyvä järjestelmätyökalu helpottamaan Windows-palvelimen ja aktiivihakemiston hallintaa etänä ja paikallisesti. Lisäksi havaittiin, että suurin hyöty laajassa ympäristössä saadaan luomalla ja suorittamalla skriptejä. Lähteinä on käytetty kirjallisia lähteitä sekä verkkojulkaisuja.

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree programme in Information and Communication Technologies
Telecommunications and Networks

IIRO KOVISTO:

Windows Server 2012 and AD –management with PowerShell

Bachelor's thesis 26 pages

June 2014

The purpose of this Bachelor thesis was to clarify PowerShell use of opportunities in Microsoft Windows Server 2012 -operating system and in active directory management. Server and active directory management in this thesis is mainly executed by making remote control outside of domain from workstation to server.

PowerShell is Microsoft command line tool for Windows operating system management. Its purpose is to make easier and optimize the system management locally and remotely. In this thesis PowerShell command line prompt tool is generally introduced, is familiarized of its history, interface, use, ISE –scripting management tool and most important commands.

Windows Server 2012 is the sixth release of server structure which Microsoft has made. This thesis presents the main features of server, the history, roles and user interfaces. Furthermore active directory usage and preferences has examined.

The practical part includes the installation of the server operating system and making the remote connection by PowerShell. Regarding the server roles management and domain creating there is also handle organizational units and user account management in active directory.

The result of this bachelor's thesis was that PowerShell tool is a good system to facilitate the Windows server and active directory management remotely and locally. The greatest benefit is received by creating and executing scripts in large environment. Written sources and online publications have been used by sources.

Key words: PowerShell, Active Directory, Windows Server 2012, Remote control

SISÄLLYS

1	JOHDANTO.....	6
2	POWERSHELL -TYÖKALU	7
2.1	Powershell versiot.....	7
2.2	Käyttöliittymä	8
2.3	Cmdlet -komennot	9
2.4	Aliakset, funktiot ja parametrit	9
2.5	Profiilit	10
2.6	Komentojen putkittaminen.....	10
2.7	Peruskomentoja.....	11
2.8	ISE (Integrated Scripting Enviroment) -skriptienhallintatyökalu.....	12
3	MICROSOFT WINDOWS SERVER 2012 - PALVELINKÄYTTÖJÄRJESTELMÄ.....	14
3.1	Versiot.....	14
3.2	Core ja GUI -käyttöliittymät.....	15
3.3	Palvelinroolit ja ominaisuudet	15
3.4	Aktiivihakemisto.....	15
4	PALVELIMEN JA AKTIIVIHAKEMISTON POWERSHELLILLÄ	16
4.1	Windows Server 2012 -Palvelinkäyttöjärjestelmän asennus	16
4.2	Etäyhteyden muodostaminen PowerShellillä	16
4.3	Palvelimen roolien hallinta ja käyttöönotto	18
4.4	Toimialueen luonti.....	19
4.5	Aktiivihakemisto -moduulin asennus PowerShell -istuntoon.....	20
4.6	Organisaatioyksikön luominen	20
4.7	Konetilien lisääminen	21
4.8	Käyttäjätilien hallinta.....	21
4.9	Käyttöoikeusryhmien hallinta.....	23
5	POHDINTA.....	24
	LÄHTEET	25

LYHENTEET JA TERMIT

AD	Active Directory, Aktiivihakemisto, Windows-toimialueen käyttäjätietokanta ja hakemistopalvelu
Alias	Powershellissä käytettävien cmdletien lyhenne
Core	Windows Server 2012 –käyttöliittymä
CMDLET	Command-Let, Powershellissä käytettäviä komentoja kutsutaan CMDLET:ksi
Funktio	Istuntoon sisäänrakennettu skripti tai skriptin osa
GUI	Windows Server 2012 –käyttöliittymä
ISE	Integrated Scripting Enviroment, PowerShellin skriptienhallintatyökalu
Parametrit	Komennolle suorituksen yhteydessä välitettävä tieto
PowerShell	Komentorivipohjainen työkalu Microsoftin Windows -käyttöjärjestelmien hallintaan
Provider	Tarjoaa pääsyn PowerShellissä Windowsin osiin
OU	Organisation Unit, Organisaatioyksikkö, AD:n ominaisuus, jolla toimialueen objekteja voidaan järjestää loogisiin kokonaisuuksiin

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena oli selvittää Powershell -työkalun käyttömahdollisuuksia Windows Server 2012 –palvelinkäyttöjärjestelmän ja aktiivihakemiston hallinnassa. PowerShell on Microsoftin komentorivipohjainen työkalu Windows –käyttöjärjestelmien hallintaan, jonka tehtävänä on helpottaa ja tehostaa järjestelmähallintaa paikallisesti ja etänä.

Työn teoriaosuudessa esitellään yleisesti PowerShell –komentotyökalu, perehdytään työkalun historiaan, käyttöliittymään, käyttökohteisiin, ISE –skriptienhallintatyökaluun ja tärkeimpiin komentoihin. Teoriaosuudessa on esitelty myös Microsoftin Windows Server 2012 -palvelinkäyttöjärjestelmän keskeisimmät ominaisuudet, historia, roolit ja käyttöliittymät. Lisäksi perehdyttiin aktiivihakemiston käyttöön ja ominaisuuksiin.

Käytännön osuudessa asennettiin palvelimelle PowerShellillä hallinnoitava käyttöjärjestelmä. Palvelimen ja aktiivihakemiston hallinta työssä suoritettiin muodostamalla etäyhteys PowerShellillä toimialueen ulkopuolisesta koneesta palvelimelle. Palvelinroolien hallinnan ja toimialueen luomisen lisäksi osuudessa käsiteltiin organisaatioyksiköiden ja käyttäjätilien hallintaa aktiivihakemistossa.

2 POWERSHELL -TYÖKALU

PowerShell on Microsoftin komentorivipohjainen työkalu Windows – käyttöjärjestelmien hallintaan. Sen tehtävänä on helpottaa ja tehostaa järjestelmän ja ohjelmistojen hallintaa paikallisesti ja etänä. PowerShelliin voidaan syöttää yksittäisiä komentoja tai luoda pidempiä komentojonoja putkittamalla komentoja. PowerShellillä on myös mahdollista luoda skriptejä, jotka suorittavat automatisoituja toimintoja. (Lee, Mitschke, Schill & Tanasovski 2011, 5-7)

2.1 Powershell versiot

Windows PowerShell:stä on julkaistu versiot 1.0, 2.0, 3.0 ja viimeisimpänä 4.0. PowerShellin ensimmäinen version 1.0 julkaistiin vuonna 2006 ja se piti sisällään hieman yli 100 erilaista cmdlet:iä. Tuetut käyttöjärjestelmät olivat Windows XP SP2, Windows Server 2003 ja Windows Vista. PowerShellin versiot on ladattavissa Microsoftin sivuilta. (PowerShell Team 2006.)

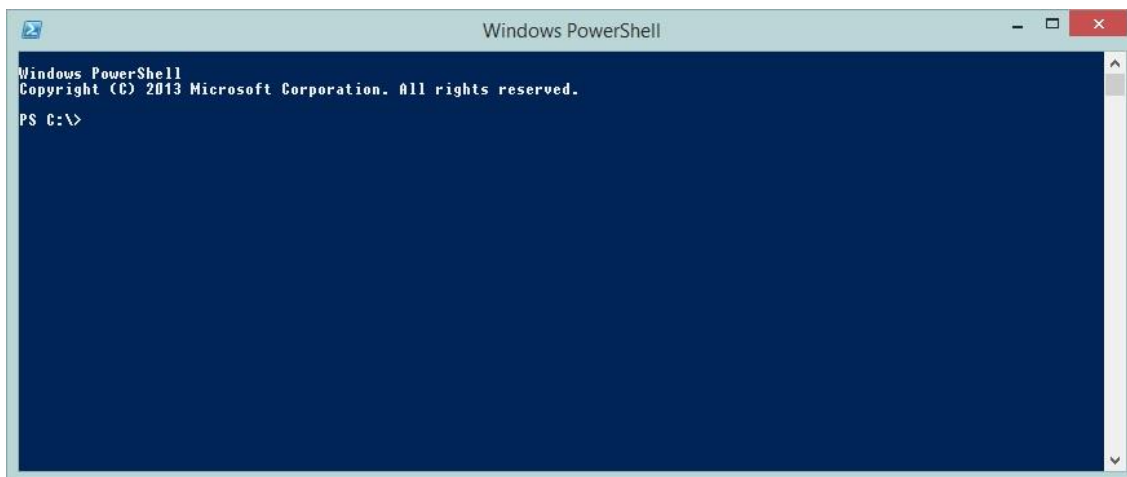
Powershell:in 2.0 -versio julkaistiin Windows 7- ja Windows Server 2008 R2 - käyttöjärjestelmien yhteydessä lokakuussa 2009. Versio 2.0:n mukana tuli runsaasti uudistuksia. Tärkeimpänä uudistuksena oli cmdlet:ien määrän tuplaantuminen ja ISE (Integrated Scripting Environment) skriptienhallintatyökalun tuleminen. Versio 2.0:n mukana myös etähallinta kehittyi. Uutena ominaisuutena tulivat myös Job –komennot, jotka mahdollistivat komentojen suorittamisen tausta-ajona. Job -komento on hyödyksi komennoissa, jotka kestävät pitkään tai jos ajetaan komentoja etänä. (Medd 2010.)

Powershell 3.0 -versio on mukana Windows 8- ja Windows Server 2012 - käyttöjärjestelmissä. Versio 3.0:n mukana tuli paljon uudistuksia. Uusien komentojen lisäksi parannuksia tuli erityisesti peruskäyttöön. ISE -skriptienhallintatyökalu päivitettiin käyttäjäystävällisemmäksi, jolloin mukaan tuotiin myös mahdollisuus ajaa komentoja ISE:n kautta. Muita uudistuksia olivat suorituskyvyn parantuminen, moduulien automaattinen lataaminen, paranneltu etähallinta, ajastettujen tehtävien luonti, PowerShell Web Access- ja Win PE -tuki. Myös PowerShellin tärkeimmän komennon, *get-help*:in tiedostojen päivittäminen helpottui. Komennolla *update-help* pystytään päivittämään koko help-kirjasto ajantasalle. (Nelson 2012.)

Powershell 4.0 -versio on mukana Windows 8.1- ja Windows Server 2012 R2 -käyttöjärjestelmissä. Pääosin 4.0 -version mukana tuli vain pieniä parannuksia ja muutamia uusia komentoja. Tärkeimpänä uusista komennoista mukaan tuli Save-Help, joka mahdollistaa ohjeiden tallentamisen etäkoneeseen asennetuista moduuleista. Uutena ominaisuutena mukaan tuli myös Windows PowerShell Desired State Configuration (DSC). DSC on hallintajärjestelmä, joka mahdollistaa ohjelmistojen, sekä palveluiden käyttöönnoton ja hallinnan muokkaamisen. (Microsoft TechNet 2014a.)

2.2 Käyttöliittymä

PowerShellin käyttöliittymä, eli konsoli, toimii sinisellä pohjalla. Komentorivillä liikutaan nuolinäppäimillä ja Ctrl + nuoli komennolla pääsee hyppäämään suoraan välilyönteihin. F7 antaa listan aiemmin kirjoitetuista komennoista, jotka suoritetaan enterillä. Tarvittaessa komennon voi kopioida nuolella komentoriville. Saman listauksen saa näkyviin konsoliin myös komennolla get-history. Tabulaattori toimii työkaluna, jonka avulla pystytään täydentämään komennot, parametrit ja muut tunnetut asiat. Taaksepäin tabulaattorilla täydennettävien listaa voi selailla shift + tabulaattori yhdistelmällä.



KUVA 1. Powershellin käyttöliittymä.

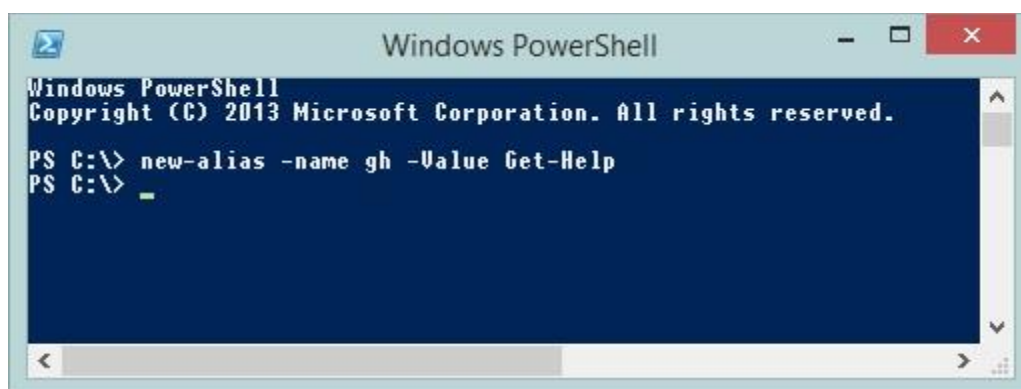
PowerShell -istunnon pystyy myös muokkamaan oman näköisekseen. PowerShelliin voidaan luoda käynnistykseen yhteydessä ajettavia käyttäjäprofiileja, jotka sisältävät aiemmin tehtyjä skriptejä. Powershellillä on mahdollista luoda myös omia funktioita ja aliaksia. (Perez 2012.)

2.3 Cmdlet -komennot

PowerShellissä käytettävät komennot koostuvat Command-Let:eistä, joista Microsoft käyttää nimitystä Cmdlet. Cmdletit koostuvat aina kahdesta osasta, verbistä ja substantiivista (esim. get-help, start-process). Kaikki PowerShellin käytössä olevat verbit saa listattua komennolla get-verb. Vastaavasti kaikki käytössä olevat cmdletit komennolla get-command. (Driscoll 2012, 5-6.)

2.4 Aliakset, funktiot ja parametrit

Cmdletien pituudesta johtuen PowerShellissä on käytössä lyhenteitä eli aliaksia. Listan käytössä olevista aliaksista saa get-alias komennolla. Monet DOS-ajoilta tutut komennot, kuten cd, dir, cls, sekä myös useat Unix komennot, kuten ls ja man toimivat PowerShellissä näiden kautta. Esimerkiksi *cd* -komento on alias *Set-Location* -cmdlet:istä. (Microsoft Technet 2014b)



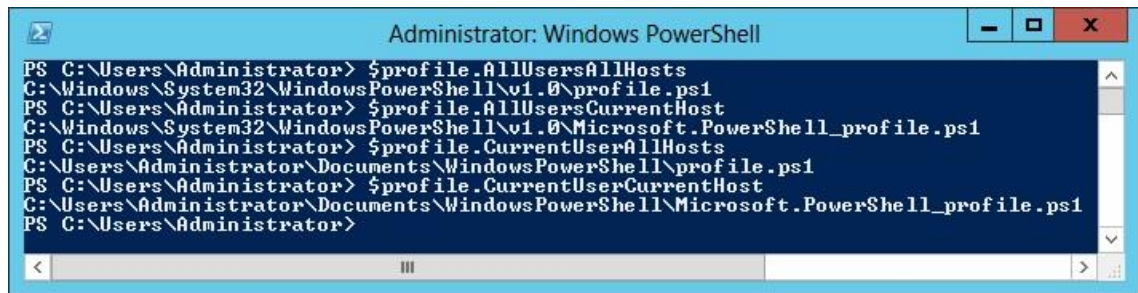
KUVA 2. Luodaan gh niminen alias, joka suorittaa get-help –cmdlet:in.

Funktiot ovat PowerShellin sisäänrakennettuja skriptejä, jotka suorittavat ennalta määrättyjä toimintoja. Istuntoon määritellyt funktiot löytyvät Function: -providerin takaa. (Jones 2011.)

PowerShellissä Cmdlet :ien yhteydessä käytetään lähes poikkeuksetta parametreja. Näiden edessä on aina '-' merkki. Myös parametrien kanssa tabulaattori toimii työkaluna. Mikäli cmdlet:lle tarvittavia parametreja ei anneta komennon yhteydessä, konsoli pyytää nämä erikseen täydentämään. (Driscoll 2012, 9.)

2.5 Profiilit

Uutta PowerShell istuntoa käynnistettäessä oletusarvoisesti ei aiemmin luotuja aliaksia ja skriptejä ole mukana. Mikäli edellämainittuja toimintoja halutaan mukaan, tulee PowerShellin sisään luoda profiili. Profiilien tyyppejä on 4. Profiilit voidaan määrittää vaikuttamaan kaikkiin käyttäjiin ja istuntoihin tai vaihtoehtoisesti vain tietyille käyttäjille. Istunnossa käytettävän profiilin saa selville *\$profile* –providerilla.

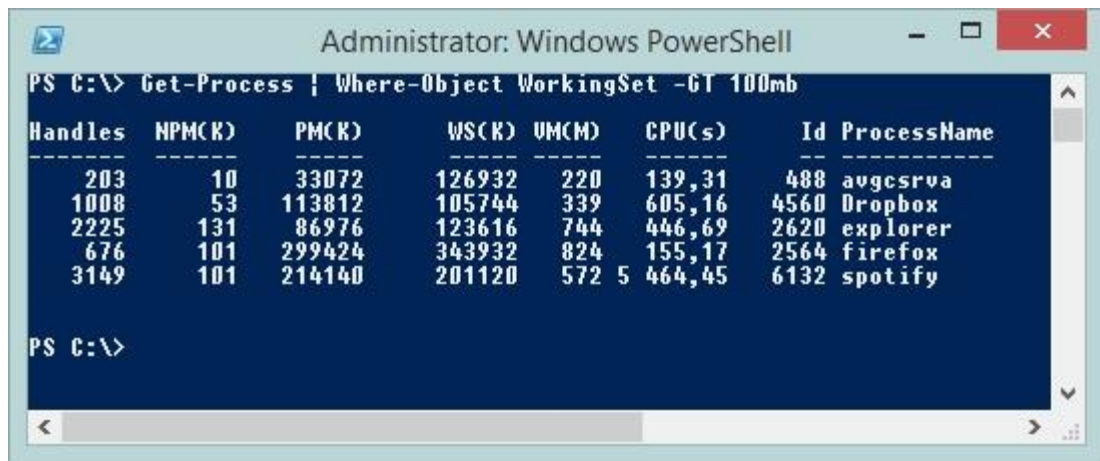


KUVA 3. Powershellin profiilityypit ja niiden sijainnit.

Profiilit ovat .ps1 päätteisiä tiedostoja ja niiden sisään voidaan määrittää komentoja, jotka suoritetaan heti istuntoa avattaessa. Profiilin sisällä voidaan suorittaa suoria komentoja, aliaksia ja funktioita. Toisin sanoen profiilit ovat PowerShellin skriptejä. (Microsoft TechNet 2009.)

2.6 Komentojen putkittaminen

Powershellin avulla on mahdollista putkittaa (piping) komentoja. Putkittaminen tarkoittaa komennon objektin arvon ottamista toisen komennon käsittelyyn. Tämä mahdollistaa pidempien komentojonojen luomisen ja niiden ketjuttamisen keskenään. Putkittamisen avulla voidaan esimerkiksi luoda taulukoita ja listauksia. Putkessa komennot erotetaan '|' –merkillä. Komentojen suoritusjärjestys voidaan määrittää matematiikasta tutuilla sulkumerkeillä '()' . (Microsoft TechNet 2014c.)

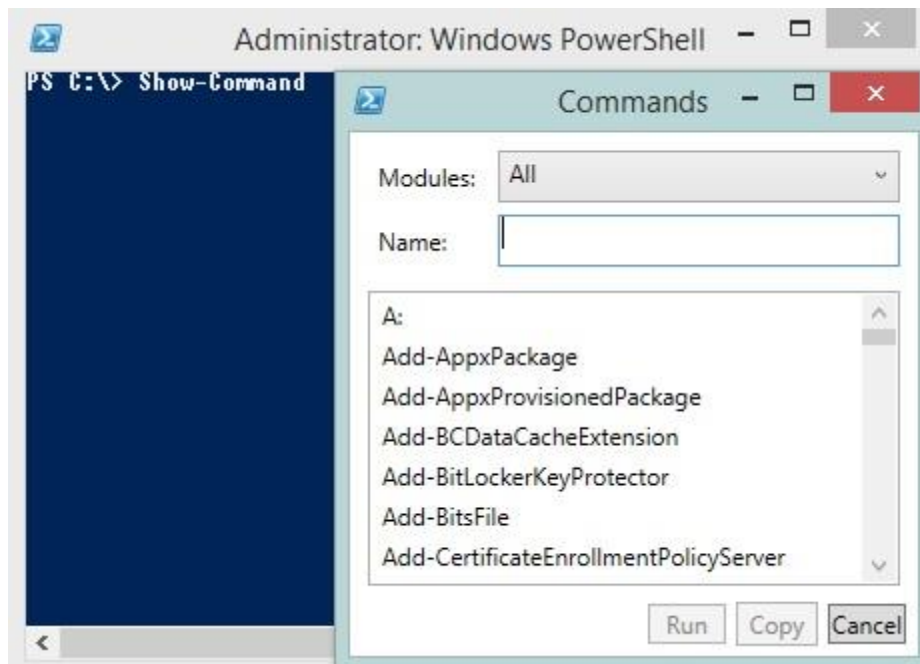


KUVA 4. Putkitettu komento, joka antaa tiedot yli 100mb muistia vievistä aktiivisista prosesseista (WS(K) -sarake).

2.7 Peruskomentoja

Get-Help -cmdlet sisältää kuvauksen kaikista PowerShellin tukemista komennoista ja näin ollen sitä pidetään yleisesti PowerShellin tärkeimpänä komentona. PowerShellissä on laaja help-kirjasto, joka on kuitenkin versio 3.0:sta lähtien oletuksena tyhjä. Koko help-kirjaston pystyy kuitenkin päivittämään Microsoftin palvelimilta komennolla Update-Help. Get-Help -komennon antamat ohjeet sekä esimerkit löytyvät myös Microsoftin sivuilta, joihin PowerShell istunto antaa kommentoa suorittaessa suoran linkin. (Driscoll 2012, 11)

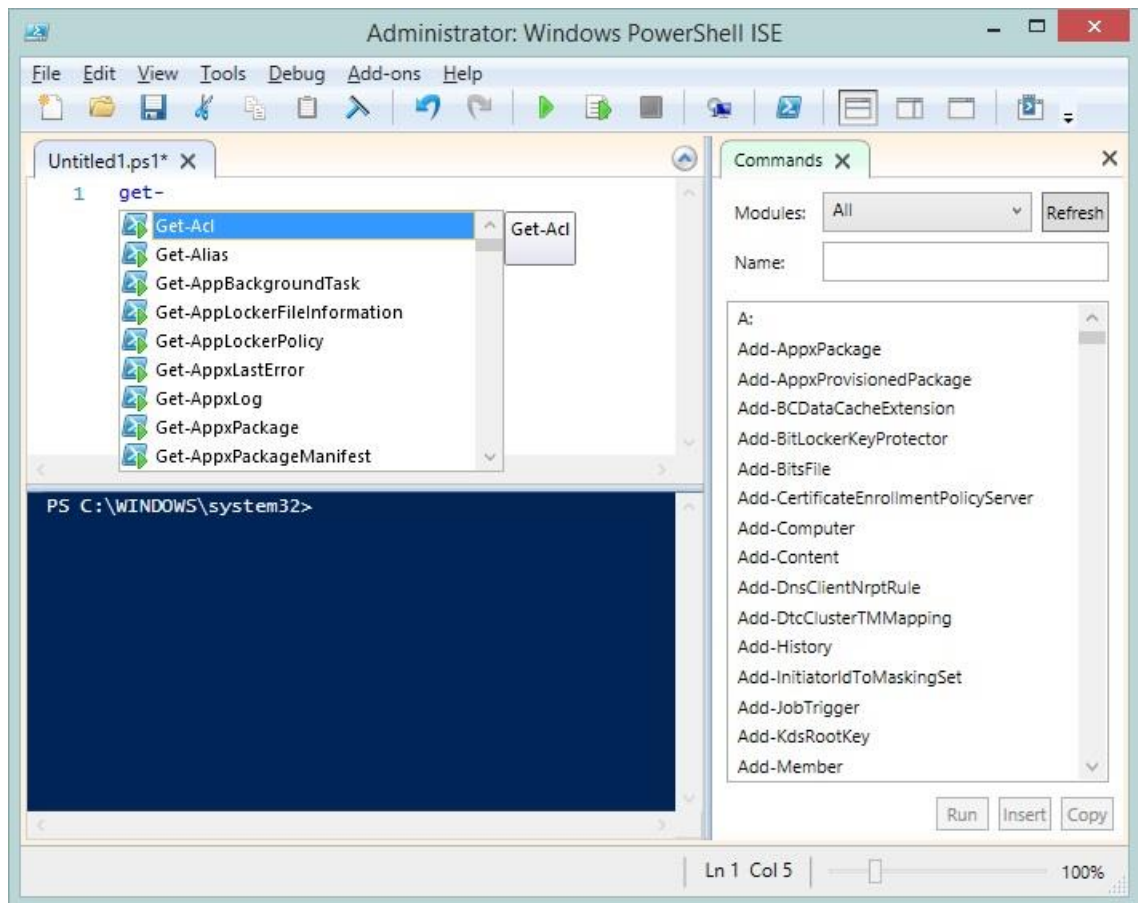
Muita tärkeitä komentoja ovat Get-Command, joka listaa käytettävät cmdlet:it ja Get-Member, joka listaa cmdletien ominaisuudet ja käytettävät parametrit. Tärkeänä komentona voidaan myös pitää Show-command -komentoa, joka on mukana PowerShellin versio 3.0 :sta lähtien. Komento antaa graafisen listauksen käytettävistä komennoista. (Driscoll 2012, 15)



KUVA 5. Show-command -komennon antama graafinen listaus suoritettavista komennoista.

2.8 ISE (Integrated Scripting Enviroment) -skriptienhallintatyökalu

PowerShellin ISE on Windowsin graafisella käyttöliittymällä toimiva skriptienluontityökalu, joka on ollut mukana PowerShellin versio 2.0:sta lähtien. Sen avulla on mahdollista luoda ja testata skriptien ja modulien toimintaa omassa ympäristössään. ISE sisältää monirivisen editointityökalun, josta koko skriptin voi suoraan suorittaa PowerShellin komentoriville. Tärkeimpinä ominaisuuksina ovat syntaksien väritys, välilehdet, visuaalinen virheenkorjaus sekä tilannekohtaisten ohjeiden saatavuus. (Microsoft TechNet, 2013.)



KUVA 6. PowerShell ISE -käyttöliittymä versiossa 4.0

3 MICROSOFT WINDOWS SERVER 2012 - PALVELINKÄYTTÖJÄRJESTELMÄ

Windows Server 2012 julkaistiin elokuussa 2012. Se on Microsoftin kuudes julkaisu kehittämästään palvelinjärjestelmästä. Valittavissa on 4 erilaista julkaisua: Foundation, Essentials, Standard, Datacenter. (Minasi, Greene, Booth, Butler, McCabe, Panek, Rice & Roth 2013, 1-2)

3.1 Versiot

Windows Server 2012:n mukana oli mahdollisuus valita kahden julkaisun, Standard:in ja Datacenter:in, välillä. Server 2012 R2:n myötä tuli kaksi uudempaa julkaisua: Foundation ja Essentials. Käytettävä versio tulee valita käyttökohteen mukaan, sillä kullakin julkaisulla on omat rajoitteensa.

Standard Edition on suunniteltu suuryrityksen pilvipalvelimeksi. Se on julkaistuista jakeluversioista suosituin. Standard Edition sisältää lähes kaikki palvelut mitä muutkin julkaisut ilman suurempia rajoitteita. Datacenter Edition sisältää kaikki samat ominaisuudet kuin Standard Edition. Sen ainoana erona on, että virtualisointien määrää ei ole rajoitettu. Datacenter Editionin hinta on lähes nelinkertainen verrattuna Standardiin.

Foundation Edition pitää sisällään suurimman osan muiden julkaisujen sisällöistä, mutta se sisältää runsaasti rajoitteita. Käyttäjää voi olla korkeintaan 15, muistin määrä on rajoitettu, sekä virtualisointi on poistettu käytöstä. Muut rajoitteet koskevat aktiivisten yhteyksien määrää.

Essentials Editionissa on hieman vähemmän rajoitteita kuin Foundationissa. Se on suunniteltu pienyrityksille, joissa käyttäjiä on alle 25 ja verkkolaitetta alle 50. Muistin määrä on kaksinkertainen verrattuna Foundationiin. Lisäksi käyttäjämäärä on suurempi, aktiivisia yhteyksiä voi olla hieman enemmän ja virtualisointioikeuksia on yksi kappale. (Minasi ym. 2013, 2-3)

3.2 Core ja GUI -käyttöliittymät

Windows Server 2012 –käyttöjärjestelmästä on tarjolla kahta versiota. Palvelimen asennusvaiheessa valitaan halutaanko käyttää Core- vai GUI -käyttöliittymää. Core sisältää ainoastaan komentorivipohjaisen käyttöliittymän. GUI –käyttöliittymässä näkymä on vastaavanlainen kuin Windows 8 -käyttöjärjestelmässä. Core ja GUI –käyttöliittymien välillä voidaan vaihtaa myös palvelimen asennuksen jälkeen asentamalla tai poistamalla GUI –palvelu käytöstä. (Hester & Henley, 2013, 9)

3.3 Palvelinroolit ja ominaisuudet

Palvelimen tarjoamia palveluita kutsutaan rooleiksi. Yleensä palvelin varataan suorittamaan vain jotain tiettyä roolia varten, mutta palvelin on myös mahdollista varata suorittamaan useampia rooleja samanaikaisesti.

Windows Server 2012 –käyttöjärjestelmä sisältää seuraavia palveluita:

- Aktiivihakemisto (AD)
- Etähallintatyökalut
- Hyper-V –virtualisointityökalu
- DHCP- ja DNS -palvelut
- Tiedosto- & tulostinpalvelin
- Verkkokäytäntöjen hallinta
- Web palvelin (IIS)

(Hester & Henley, 2013, 7-9)

3.4 Aktiivihakemisto

Aktiivihakemisto (Active Directory) on Microsoftin kehittämä organisaatioyksiköiden, käyttäjätilien ja verkkolaitteiden hallintaan tarkoitettu järjestelmätyökalu. AD sisältää tietokannan käyttäjätileistä sekä verkon laitteista ja määrittelee näiden pääsyn verkon sovelluksiin ja palveluihin. Lisäksi se tarjoaa hakemistopalvelut, jotka mahdollistavat verkon infrastruktuurin määrittelemisen ja turvaamisen. (Hester & Henley 2013. 45)

4 PALVELIMEN JA AKTIIVIHAKEMISTON HALLINTA POWERSHELLILLÄ

Tässä luvussa on käsitelty Windows Server 2012 –palvelinkäyttöjärjestelmän asennus, etäyhteyden muodostaminen PowerShellillä, sekä aktiivihakemiston hallinta. Luvussa on käsitelty myös palvelimen roolien hallintaa ja käyttöönottoa. Aktiivihakemiston hallinnassa on toimialueen luomisen lisäksi käsitelty organisaatioyksiköiden, käyttäjätilien, konetilien ja käyttöoikeusryhmien hallinta. Osuuden teossa on käytetty PowerShellin versiota 4.0.

4.1 Windows Server 2012 -Palvelinkäyttöjärjestelmän asennus

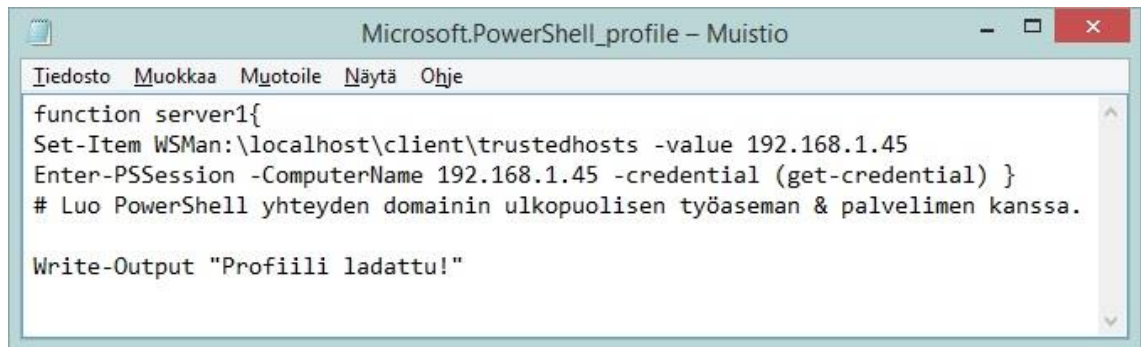
Työssä käytettiin Windows Server 2012 -käyttöjärjestelmästä Standard –jakelua. Asennusmediat ladattiin Microsoftin Dreamspark –palvelusta, johon tunnukset saatiin Tampereen Ammattikorkeakoulun puolesta. Asennus tehtiin kokonaan erilliselle työasemalle.

Server 2012 Standard –jakelun asennus oli nopea ja yksinkertainen. Levyjakojen lisäksi ainoa merkittävä käyttäjän tekemä valinta oli GUI- ja Core -käyttöliittymien välillä, josta mainittiin aiemmin luvussa 3.2. Työssä valittiin käytettäväksi graafinen GUI -käyttöliittymä.

4.2 Etäyhteyden muodostaminen PowerShellillä

Koska työn tarkoituksena oli hallita palvelinta PowerShellillä, luotiin etäyhteys PowerShell -istunnon kautta. Ainoa asennuksen jälkeinen muutos palvelimen päässä oli etäyhteyksien asettaminen sallituiksi. PowerShell -etäistunnon luomista varten käynnistettiin etäkoneesta WinRM -palvelu ja lisättiin palvelimen osoite luotetuksi komennolla *Set-Item WSMAN:\localhost\client\trustedhosts -value "Palvelimen ip-osoite"*. Yhteys muodostettiin komennolla *Enter-PSSession -ComputerName "Palvelimen ip-osoite" -credential (get-credential)*.

Etäyhteys palvelimelle luotiin toimialueen ulkopuolisesta koneesta. Yhteyden ottamisen helpottamiseksi työaseman PowerShellin profiiliin luotiin funktio server1, joka piti sisällään molemmat yllä mainitut etäyhteyden luomiseen vaadittavat komennot. Funktion tehtävänä oli mahdollistaa etäyhteyden ottaminen palvelimeen kahden komentokehoterivin sijaan yhdellä sanalla.

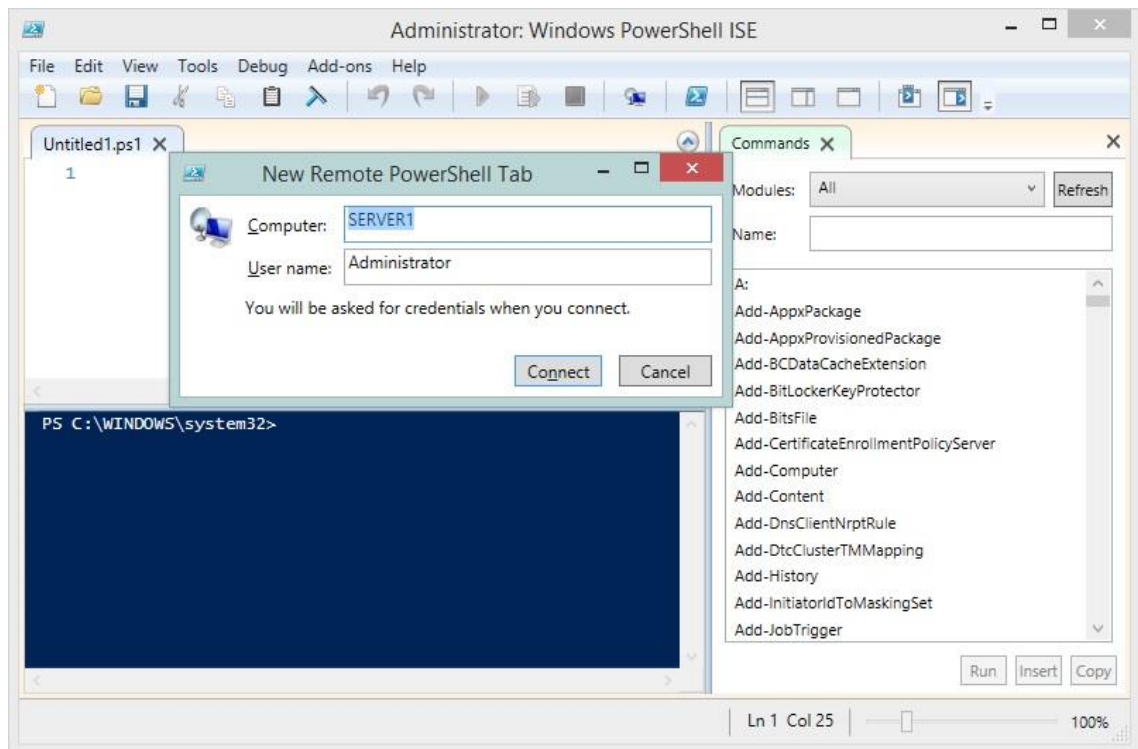


KUVA 7. Funktion server1 lisääminen profiiliin.



KUVA 8. PowerShell konsoliyhteyden muodostaminen etäkoneelta palvelimelle.

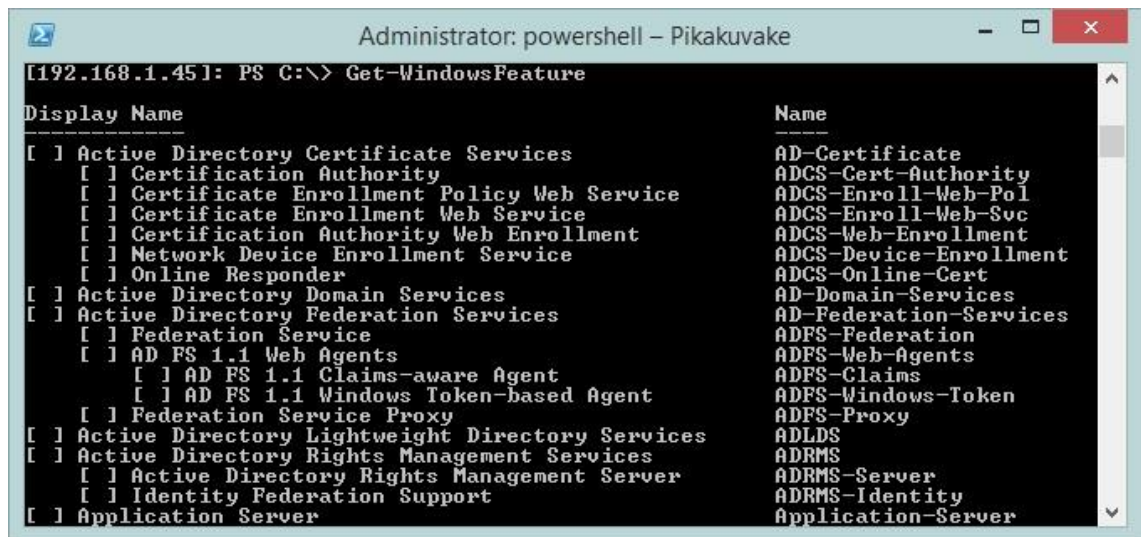
Etäyhteys voidaan muodostaa myös ISE –työkalulla. ISE:n File –välilehdeltä valitaan New Remote PowerShell Tab, jonka jälkeen syötetään palvelimen nimi ja kirjautumistunnus. Yhteyden muodostuttua istunto avautuu uuteen välilehteen.



KUVA 8. Etäyhteyden muodostaminen ISE -työkalulla.

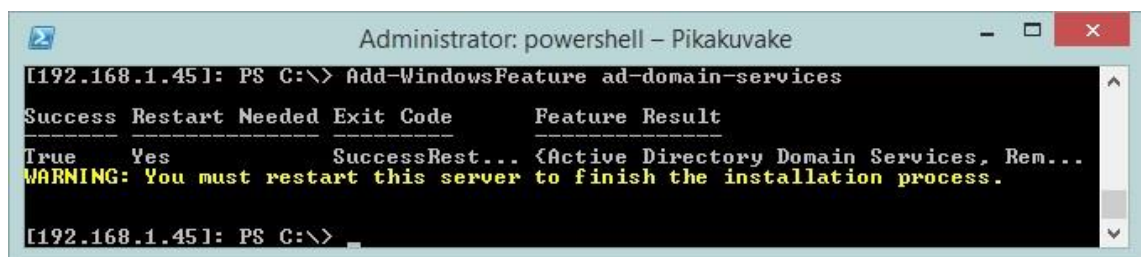
4.3 Palvelimen roolien hallinta ja käyttöönotto

Lista Windows Server 2012 –käyttöjärjestelmälle asennetuista ja asennettavissa olevista rooleista saadaan listattua komennolla *Get-WindowsFeature*. Kuvassa 9. on esitetty komennon tulostama puumainen listaus. Lista kertoo roolin lisäksi myös lisättävän roolin PowerShell -komennon, jotka saadaan käyttöön *New-WindowsFeature* -komennolla. Aktiivihakemistoroolit eivät ole oletuksena palvelimen asennuksen jälkeen aktiivisina, joten ne täytyy ottaa erikseen käyttöön. Kaikki PowerShellillä AD:n hallinnassa käytettävissä olevat cmdlet:it saadaan listattua komennolla *Get-Command -Module ActiveDirectory*.



KUVA 9. Palvelimelle asennetut ja asennettavissa olevat roolit.

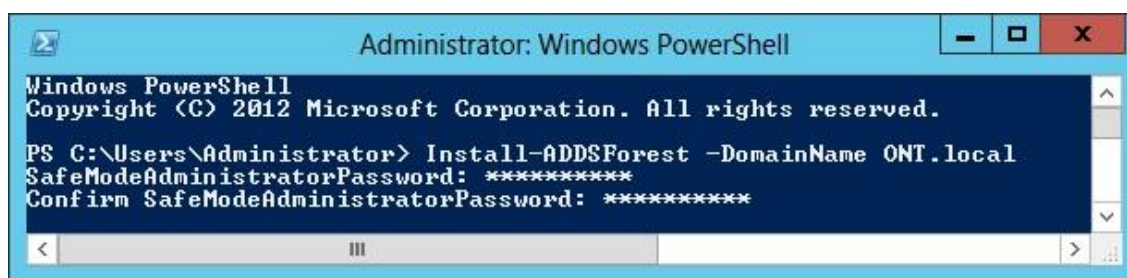
Komennolla `ADD-windowsfeature AD-Domain-Services` otettiin käyttöön aktiivihakemiston toimialuepalvelu –rooli. (Active Directory Domain Service). Lähes kaikkien roolien käyttöönottamisen tai käytöstä poistamisen jälkeen palvelin täytyy käynnistää uudelleen. Uudelleenkäynnistys onnistuu komennolla `Restart-Computer`.



KUVA 10. Aktiivihakemiston toimialue -roolin ottaminen käyttöön.

4.4 Toimialueen luonti

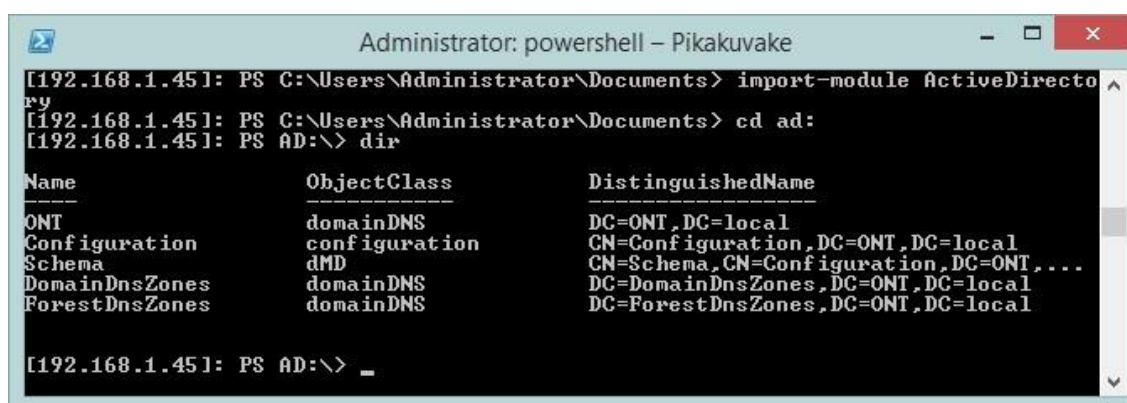
Aktiivihakemisto –roolin käyttöönoton jälkeen luotiin palvelimelle toimialue. Uutta toimialuetta luodessa asennettiin ensimmäiseksi aktiivihakemiston toimialuepalveluksi metsä –palvelu. Komennon syöttämisen yhteydessä annettiin myös toimialueelle nimi (ONT.local). Toimialueen luonti suoritettiin komennolla `Install-ADDSTForest -DomainName ONT.local`, jonka jälkeen palvelin vaati uudelleen käynnistys. Toimialueen luomisen yhteydessä määritettiin aktiivihakemistolle salasana.



KUVA 11. Toimialueen ONT.Local luonti PowerShellillä.

4.5 Aktiivihakemisto -moduulin asennus PowerShell -istuntoon

Toimialueen luomisen jälkeen asennettiin aktiivihakemiston hallintaan tarvittava Active Directory –moduuli komennolla *Import-Module ActiveDirectory*. Moduuli täytyy lisätä aina uuden istunnon yhteydessä. Vaihtoehtoisesti komennon voi myös lisätä profiiliin. Moduulin lisäämisen jälkeen siirryttiin PowerShellin sisällä AD:hen. Aktiivihakemistossa liikkuminen PowerShellillä tapahtui samoin kuin normaalissa kansiorakenteessa.



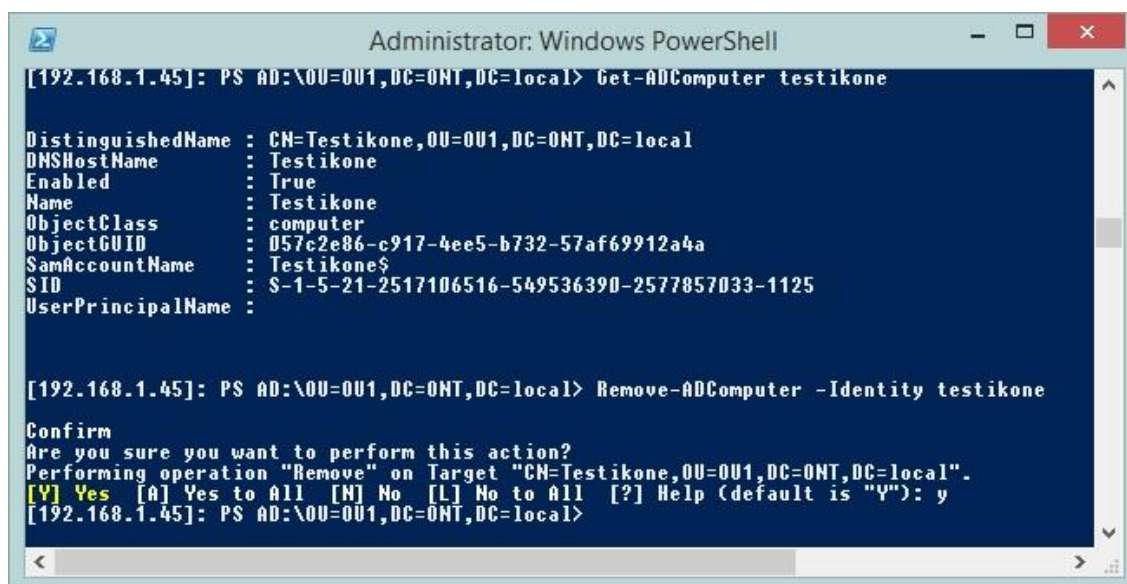
KUVA 12. Siirtyminen domainin ulkopuoliselta koneelta palvelimen aktiivihakemistoon.

4.6 Organisaatioyksikön luominen

Uusi organisaatioyksikkö luodaan cmdlet:illä *New-ADOrganizationalUnit*. Parametreiksi tulee määritellä *-name* ja *-Path*. Organisaatioyksiköitä voidaan muodostaa myös jo olemassa olevien OU:iden sisään muuttamalla parametrin *-path* polkua vastaamaan olemassa olevaa organisaatioyksikköä. Organisaatioyksikkö luotiin komennolla *New-ADOrganizationalUnit -name OUI -Path '\DC=ONT,DC=local'*.

4.7 Konetilien lisääminen

Uusi konetili luodaan cmdlet:illä *New-ADComputer*. Parametrina tulee antaa vähintään toimialueelle lisättävän koneen nimi. Konetilin tiedot saadaan cmdlet:illä *Get-ADComputer* ja lisäämällä komentoon luodun konetilin nimi. Konetilin poisto onnistuu *Remove-ADComputer* komennolla.

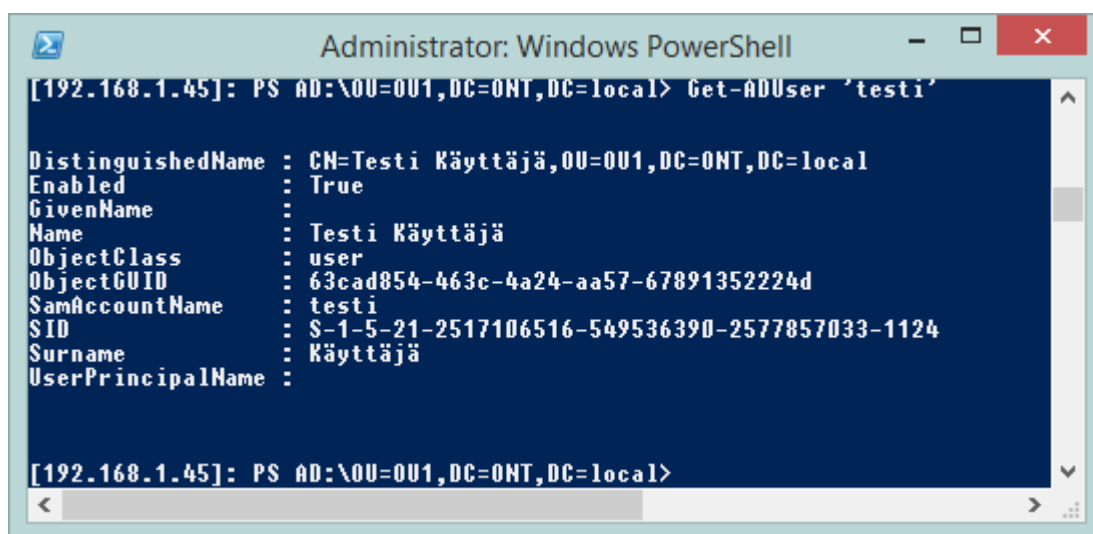


KUVA 13. Konetilin tietojen tulostaminen PowerShellillä ja poistaminen toimialueelta.

4.8 Käyttäjätilien hallinta

Uusien käyttäjätilien luominen aktiivihakemistoon onnistuu komennolla *New-ADUser*. Komennon parametreina voidaan antaa käyttäjätilin tietoja, kuten esimerkiksi etunimi, puhelinnumero, tilin kuvaus tai määrittää salasana. Pakollisina parametreina uuden käyttäjätilin luonnissa tulee kuitenkin antaa *Name* ja *SamAccountName*. *Name*-parametri on käyttäjätilin käyttäjänimi ja *SamAccountName* on vastaavasti kirjautumisen yhteydessä käytettävä tunnus. Yksinkertaisimmillaan käyttäjätilin luonti onnistuu siirtymällä aktiivihakemiston puussa haluttuun organisaatioyksikköön ja luomalla käyttäjätili komennolla *New-ADUser -Name 'Testi Käyttäjä' -SamAccountName 'testik'*.

Käyttäjätili on lisäämisen jälkeen oletuksena poistettu käytöstä, ellei sitä erikseen ole parametreihin määritelty. Käyttäjättilille voidaan määrittää salasana komennolla *Set-ADAccountPassword 'Testikäyttäjä' -NewPassword (ConvertTo-SecureString -AsPlainText -string 'Password2014' -force)*. Käyttäjä voidaan pakottaa vaihtamaan salasansa ensikirjautumisen yhteydessä komennolla *Set-ADUser testi -ChangePasswordAtLogon \$true*. Edellämainitut komennot voidaan myös lisätä parametreina *New-ADUser* cmdlet:in parametreihin, jolloin tili saadaan luotua yhdellä komennolla. Käyttäjätilien tarkastelu on mahdollista PowerShellillä cmdlet:llä *Get-ADUser*.



KUVA 14. *Get-ADUser* cmdlet:in antamat tiedot käyttäjätilistä.

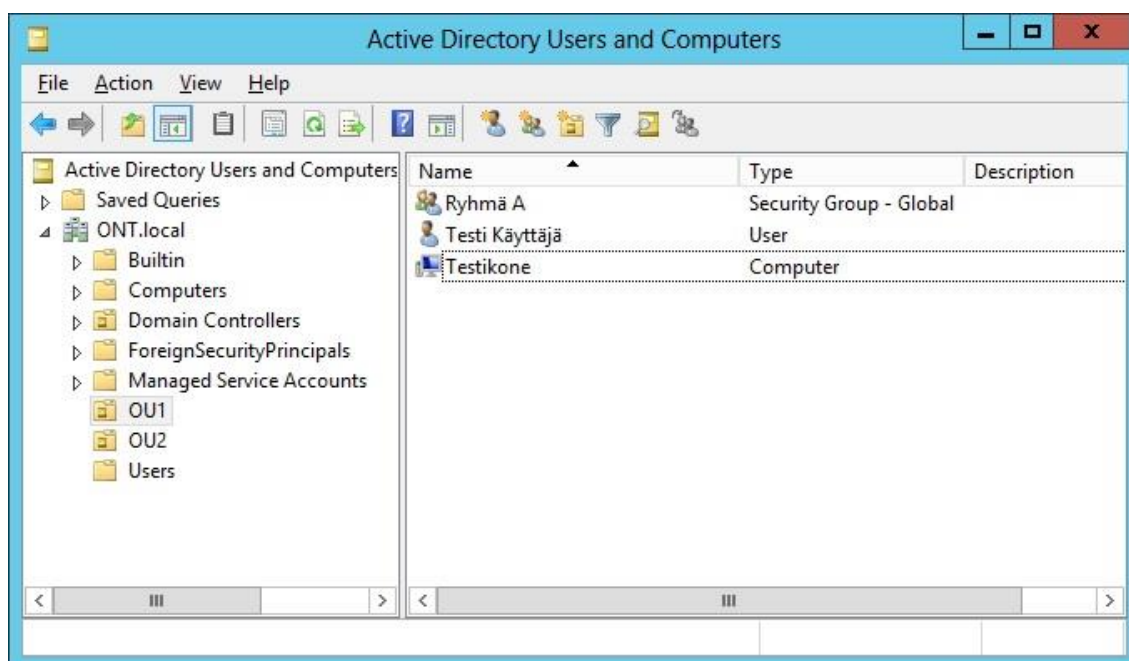
Useammalle käyttäjättilille samanaikaisesti suoritettava toiminto onnistuu suodattamalla *Filter* parametrilla jokin tilejä yhdistävä parametri ja putkittamalla haluttu toiminto. Esimerkiksi kaikkien käyttäjättilien ottaminen käyttöön, joille on määritelty organisaatioksi Tamk, suoritetaan komennolla *Get-ADUser -Filter "Organization -eq 'tamk'" | Enable-ADAccount*. Käyttäjättilin poisto tapahtuu *Remove-ADUser* -cmdlet:llä.

Aktiivihakemiston objektien siirto onnistuu *Move-ADObject* cmdlet:illä. Parametreiksi tulee antaa vähintään *Identify*, jolla objektin nimi määritellään sekä *TargetPath*, joka sisältää siirrettävän objektin kohteen. Testikäyttäjän siirto organisaatioyksikkö OU1:stä OU2:een suoritetaan komennolla *Move-ADObject -Identity testi -TargetPath '\OU=OU2'*.

4.9 Käyttöoikeusryhmien hallinta

Uusi käyttöoikeusryhmä (security group) luodaan cmdlet:illä *New-ADGroup*. Uutta ryhmää luodessa parametreina tulee antaa vähintään ryhmän nimi, sekä ryhmän tyyppi (GroupScope). Groupsopen parametreiksi voidaan määritellä joko *DomainLocal*, *Glopal* tai *Universal*. Testiryhmä luotiin komennolla *New-ADGroup -Name 'Ryhmä A' -GroupScope DomainLocal*.

Käyttäjätilien lisääminen käyttöoikeusryhmään tehdään cmdlet:illä *Add-ADGroupMember*. Parametreina tulee määritellä vähintään *Identify* ja *SamAccountName*. *Identify* -parametrille määritellään haluttu käyttöoikeusryhmä ja *SamAccountName* -parametriksi annetaan ryhmään lisättävä käyttäjätilin kirjautumisnimi. Työssä aiemmin luodun testikäyttäjän lisääminen ryhmään suoritettiin komennolla *Add-ADGroupMember -Identity 'Ryhmä A' -Members testi*. Käyttöoikeusryhmän poisto onnistuu cmdlet:illä *Remove-ADGroup*.



Kuva 15. Aktiivihakemiston näkymä palvelimelta graafisesti.

5 POHDINTA

Opinnäytetyön tavoitteena oli selvittää Powershell -työkalun käyttömahdollisuuksia Windows Server 2012 –palvelinkäyttöjärjestelmän ja aktiivihakemiston hallinnassa. Useiden PowerShell komentojen suorittaminen onnistuu myös vaihtoehtoisilla tavoilla. Esimerkiksi aktiivihakemiston käyttäjätilien ja organisaatioyksiköiden hallinnoinnin pystyisi tekemään myös palvelimen omalla graafisella työkalulla.

PowerShellin toiminnoista on kuitenkin hyvä tietää, koska työkalulla pystytään suorittamaan useita komentoja, jotka säästävät merkittävästi resursseja. PowerShell on myös hyvä työkalu työasemien hallintaan etänä, sillä sen avulla voidaan suorittaa etäkoneen kautta kaikki samat toiminnot kuin koneelle paikallisesti kirjautuen.

Työtä tehdessä suurimmaksi haasteeksi muodostui työn rajaaminen. PowerShell -työkalu on todella kattava ja laajuuden vuoksi työssä keskityttiin ainoastaan opinnäytetyön kannalta merkittävimpiin toimintoihin. PowerShell on kuitenkin loistava lisätyökalu järjestelmäasiantuntijoille. Lisäksi sen käyttöön liittyviä ohjeita ja oppaita on verkossa tarjolla runsaasti, mitkä helpottavat työkalun käyttöönottoa.

LÄHTEET

Driscoll, A. 2012. Microsoft Windows PowerShell 3.0 First Look. GBR: Packt Publishing Ltd.

Hester, M. Henley, C. 2013. Microsoft Windows Server 2012 Administration Instant Reference. USA: Wiley.

Jones, D. 2011. Windows PowerShell: Build a Better Function. Luettu 8.5.2014.

<http://technet.microsoft.com/en-us/magazine/hh360993.aspx>

Lee, T. Mitschke, K. Schill, M. Tanasovski, T. 2011. Windows PowerShell 2.0 Bible. USA: Wiley.

Medd, J. 21.9.2010. Powershell: PowerShell Version 2: What is new and why is it important? Luettu 10.3.2014

<https://www.simple-talk.com/sysadmin/powershell/powershell-version-2-what-is-new-and-why-is-it-important/>

Microsoft Technet. 17.10.2013. Technet: Introducing the Windows PowerShell ISE. Luettu 27.3.2014

<http://technet.microsoft.com/en-us/library/dd315244.aspx>

Microsoft Technet. 6.5.2014a. Technet: What's New in Windows PowerShell. Luettu 7.5.2014 <http://technet.microsoft.com/en-us/library/hh857339.aspx>

Microsoft Technet, 2014b. Windows PowerShell Aliases. Luettu 9.5.2014

<http://technet.microsoft.com/en-us/library/ee692685.aspx>

Microsoft Technet. 2014c. Technet: Piping and the Pipeline in Windows PowerShell.

Luettu 24.4.2014 <http://technet.microsoft.com/en-us/library/ee176927.aspx>

Microsoft Technet. 20.5.2009. Technet: Windows PowerShell Profiles. Luettu

18.4.2014 <http://technet.microsoft.com/en-us/library/bb613488%28v=vs.85%29.aspx>

Microsoft. 2014. Windows Management Framework 3.0: Details. Luettu 26.3.2014
<http://www.microsoft.com/en-us/download/details.aspx?id=34595>

Minasi, M. Greene, K. Booth, C. Butler, R. McCabe, J. Panek, R. Rice, M. & Roth, S.
2013. Mastering Windows Server 2012 R2. USA: Wiley.

Nelson, B. 11.12.2012. PowerShell 3.0 Features Make it More Necessary Than Ever.
Luettu 26.3.2014 <http://blog.pluralsight.com/new-powershell-3-0-features-make-it-more-necessary-than-ever>

Perez, C. 27.3.2012. Shell is Only the Beginning: Introduction to Microsoft PowerShell
– What is it and Setup. Luettu 2.4.2014.
<http://www.darkoperator.com/blog/2012/3/28/introduction-to-microsoft-powershell-ndash-what-is-it-and-se.html>

PowerShell Team. 27.10.2006. Windows PowerShell Blog: It's a Wrap! Windows PowerShell 1.0 Released! Luettu 10.3 2014.
<http://blogs.msdn.com/b/powershell/archive/2006/11/14/windows-powershell-1-0-released.aspx>